

Many-Objective Problems Are Not Always Difficult for Pareto Dominance-Based Evolutionary Algorithms

Hisao Ishibuchi¹ and Takashi Matsumoto² and Naoki Masuyama² and Yusuke Nojima²

Abstract. Recently, it has repeatedly been reported that the search ability of Pareto dominance-based multi-objective evolutionary algorithms severely deteriorates with the increase in the number of objectives. In this paper, we examine the generality of the reported observations through computational experiments on a wide variety of test problems. First, we generate 18 types of test problems by combining various properties of Pareto fronts and feasible regions. Next, we examine the performance of a frequently-used Pareto dominance-based evolutionary algorithm called NSGA-II on the generated test problems in comparison with four decomposition-based algorithms. We observe that the performance of NSGA-II severely degrades for three types of many-objective test problems which are similar to frequently-used DTLZ1-4 test problems with triangular Pareto fronts. However, better results are obtained by NSGA-II than all the examined decomposition-based algorithms for nine types of test problems even when they have ten objectives. Then, we discuss why NSGA-II does not work well on DTLZ type test problems whereas it works well on other test problems.

1 INTRODUCTION

A multi-objective optimization problem with m objectives to be minimized can be written as follows [1]:

$$\text{Minimize } f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x}), \quad (1)$$

$$\text{subject to } \mathbf{x} \in S \subset \mathcal{R}^n, \quad (2)$$

where $f_i(\mathbf{x})$ is the i th objective function ($i = 1, 2, \dots, m$), $\mathbf{x}$ is an n -dimensional decision vector, and S is the feasible region of $\mathbf{x}$. The objective functions map a solution $\mathbf{x}$ in an n -dimensional decision space to a point $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x}))^T$ in an m -dimensional objective space. We assume that the number of decision variables is larger than the number of objectives (i.e., $n > m$).

In general, multi-objective problems do not have a single optimal solution since objectives are conflicting with each other. No single solution simultaneously optimizes all objectives. Thus, multi-objective problems have a set of solutions which are viewed as having the same optimality called “Pareto optimal”. In multi-objective optimization, solutions are compared using the following Pareto dominance relation: A solution $\mathbf{a}$ is referred to as being

dominated by another solution $\mathbf{b}$ when the following relations hold: $f_i(\mathbf{b}) \leq f_i(\mathbf{a})$ for $i = 1, 2, \dots, m$ and $f_j(\mathbf{b}) < f_j(\mathbf{a})$ for at least one j . If a solution $\mathbf{a}$ is not dominated by any other solutions, $\mathbf{a}$ is a Pareto optimal solution. The set of all Pareto optimal solutions is the Pareto optimal solution set, which is the solution set of the multi-objective problem. The set of points $\mathbf{f}(\mathbf{x})$ corresponding to all Pareto optimal solutions is called the Pareto front. That is, the Pareto front is the projection of the Pareto optimal solution set to the objective space by $\mathbf{f}(\mathbf{x})$. The Pareto front of the m -objective problem in (1)-(2) is an $(m-1)$ -dimensional manifold in the m -dimensional objective space except for the case of degenerate Pareto fronts [2]. The Pareto front shows the tradeoff relation among the m objectives in the objective space.

A large number of evolutionary multi-objective optimization (EMO) algorithms have been proposed in the literature (e.g., [3]-[16]). Each EMO algorithm is designed to search for a set of well-distributed solutions over the entire Pareto front. The obtained solution set is evaluated by its approximation quality of the Pareto front. Various quality indicators have been proposed to evaluate the overall approximation quality of the obtained solution set as well as its different aspects such as the convergence of solutions towards the Pareto front and the diversity of solutions over the Pareto front [1], [17]-[19]. The hypervolume (HV [20]) indicator and the inverted generational distance (IGD [21]) indicator have been frequently used as overall performance indicators.

EMO algorithms are often categorized into three classes based on their fitness evaluation mechanisms: Pareto dominance-based, indicator-based, and decomposition-based EMO algorithms. Pareto dominance-based algorithms such as SPEA [3] and NSGA-II [4] use the Pareto dominance relation among solutions in the current population as the primary fitness evaluation criterion. Non-dominated solutions have higher fitness values than dominated ones. A diversity-based fitness evaluation is used as a secondary criterion. Indicator-based algorithms such as SMS-EMOA [5] and HypE [7] have a performance indicator for fitness evaluation. The contribution of each solution to the performance indicator value of the current population is calculated as the fitness of the solution. In decomposition-based algorithms such as MOEA/D [6] and NSGA-III [8], a multi-objective problem is decomposed into a number of single-objective problems. Each single-objective problem has the same scalarizing function and a different weight vector. A single solution is assigned to each single-objective problem. All single-objective problems are optimized in a cooperative manner towards different directions in the objective space along the weight vectors.

Many-objective optimization has been a hot topic in the last decade in the EMO community [22]-[24]. It was reported in many studies (e.g., [25]-[28]) that the performance of Pareto dominance-

¹ Shenzhen Key Laboratory of Computational Intelligence, University Key Laboratory of Evolving Intelligent Systems of Guangdong Province, Department of Computer Science and Engineering, Southern University of Science and Technology, China, email: hisao@sustech.edu.cn (Corresponding Author: Hisao Ishibuchi)

² Department of Computer Science and Intelligent Systems, Graduate School of Engineering, Osaka Prefecture University, Sakai, Osaka 599-8531, Japan, emails: {takashi.matsumoto@ci, masuyama@, nojima@}cs.osakafu-u.ac.jp

based algorithms severely degrades with the increase in the number of objectives. This is because almost all solutions become non-dominated with each other in very early generations when a multi-objective problem has many objectives (e.g., ten objectives). If all solutions in the current population are non-dominated with each other, the same fitness value is assigned to all solutions by the Pareto dominance relation. As a result, no strong selection pressure to push the population towards the Pareto front is generated. This is the main reason for the severe performance deterioration of Pareto dominance-based algorithms with the increase in the number of objectives. Indicator-based algorithms with the HV indicator work well on many-objective problems [26]. However, due to the large computation load for calculating the HV contribution of each solution, some modifications are needed when they are applied to many-objective problems (e.g., approximate HV calculation [7], the use of a different indicator such as IGD [15]).

Recently a large number of decomposition-based algorithms have been proposed for many-objective optimization [8]-[14], [16]. They have the same (or similar) basic framework as MOEA/D [6]. Decomposition-based algorithms work well on many-objective problems since the fitness evaluation of each solution is based on a scalarizing function. Some scalarizing functions can generate a strong selection pressure towards the Pareto front even in a high-dimensional objective space. Usually the calculation of scalarizing function values is much faster than Pareto dominance-based and indicator-based fitness evaluation for many-objective problems.

Whereas NSGA-II is the most well-known and frequently-used EMO algorithm, it has not been used in computational experiments for performance comparison in recent studies on many-objective optimization [9]-[16]. NSGA-II is implicitly assumed to be clearly outperformed by new many-objective algorithms. However, this implicit assumption has not been examined on a wide variety of test problems yet. Severe performance deterioration of NSGA-II with the increase in the number of objectives was demonstrated in [28] where NSGA-II was outperformed by random search on the DTLZ2 test problem [29] with 10-20 objectives. The DTLZ1 and DTLZ2 test problems were used in the MOEA/D paper [6] to demonstrate that MOEA/D clearly has higher search ability than NSGA-II. In the NSGA-III paper [8], the DTLZ1-4 test problems were used to show that NSGA-III has higher search ability than MOEA/D. Severe performance deterioration of NSGA-II was also reported for many-objective knapsack problems in [22], [25]. However, in some studies [30], [31], it was also reported that the performance of NSGA-II does not always severely deteriorate with the increase in the number of objectives. For example, it was reported in [31] that MOEA/D and HypE were outperformed by NSGA-II on knapsack problems with 8-10 objectives when the objectives were strongly correlated. These studies suggest the possibility of NSGA-II to outperform recent decomposition-based algorithms on some other many-objective test problems whereas it does not work well on many-objective DTLZ test problems [28].

In this paper, we examine the performance of NSGA-II in comparison with four decomposition-based algorithms (MOEA/D [6], NSGA-III [8], MOEA/DD [9] and θ -DEA [10]) by generating 18 types of test problems with different shapes of Pareto fronts and feasible regions. Our experimental results demonstrate that the performance of NSGA-II does not always severely deteriorate with the increase in the number of objectives. Actually, better results are obtained by NSGA-II than all the examined decomposition-based algorithms for a half of ten-objective test problems (i.e., nine types of ten-objective test problems).

The remainder of this paper is organized as follows. In Section 2, we propose 18 types of test problems by combining different properties of the Pareto front shape and the feasible region shape. Frequently-used DTLZ2-4 [29] are classified as one of the 18 types, and WFG4-9 [32] are classified as another type. This means that the performance evaluation in recent many-objective studies [8]-[16] has been performed on a limited set of test problems. Nine types of test problems in this paper have not been examined in the literature. In Section 3, NSGA-II is compared with MOEA/D [6], NSGA-III [8], MOEA/DD [9] and θ -DEA [10] by applying them to the 18 types of test problems with 3-10 objectives. Severe performance deterioration of NSGA-II is observed only on three types of test problems which are similar to DTLZ1-4. It is also observed that NSGA-II works well on a half of test problems even when they have ten objectives. In Section 4, we discuss our experimental results. We explain why the performance of NSGA-II severely degrades on many-objective test problems similar to DTLZ1-4. We also explain why NSGA-II works well on a half of the 18 types of test problems. In Section 5, we conclude this paper.

2 TEST PROBLEMS

We generate 18 types of test problems by combining all of the following properties (i.e., $3 \times 2 \times 3$ combinations):

- (i) Curvature of the Pareto front: linear, concave and convex.
- (ii) Shape of the Pareto front: Triangular and inverted triangular.
- (iii) Shape of the feasible region in the objective space:
DTLZ type, WFG type, Minus-DTLZ type.

These three aspects (Pareto front curvature, Pareto front shape, and feasible region shape) are explained in Figs. 1-3, respectively.

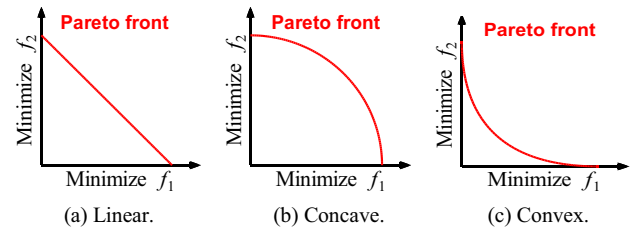


Figure 1. Three types of the Pareto front curvature.

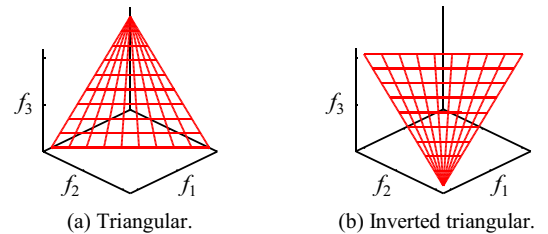


Figure 2. Two types of the Pareto front shape.

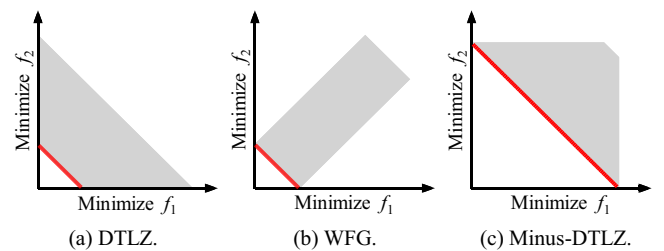


Figure 3. Three types of the feasible region shape in the objective space.

The effects of the feasible region shape on the performance of EMO algorithms have not been discussed in detail in the literature. When solutions approach the Pareto front, the size of the feasible region in the objective space decreases in DTLZ [29] in Fig. 3 (a), does not change in WFG [32] in Fig. 3 (b), and increases in Minus-DTLZ [33] in Fig. 3 (c).

Table 1 summarizes the 18 types of our test problems together with the related existing test problems. The frequently-used many-objective DTLZ2-4 and WFG4-9 test problems have concave triangular Pareto fronts (types 07 and 08 in Table 1). From Table 1, we can see that the comparison of many-objective algorithms has been performed on a limited set of test problems in the literature.

Table 1. Summary of 18 types of our test problems.

Type	Curvature	Pareto Front	Feasible Region	Related Problems
01	Linear	Triangular	DTLZ	DTLZ1 [29]
02	Linear	Triangular	WFG	-
03	Linear	Triangular	Minus-DTLZ	-
04	Linear	Inverted	DTLZ	Inverted DTLZ1 [34]
05	Linear	Inverted	WFG	-
06	Linear	Inverted	Minus-DTLZ	Minus-DTLZ1 [33]
07	Concave	Triangular	DTLZ	DTLZ2-4 [29]
08	Concave	Triangular	WFG	WFG4-9 [32]
09	Concave	Triangular	Minus-DTLZ	-
10	Concave	Inverted	DTLZ	Inverted DTLZ2 [34]
11	Concave	Inverted	WFG	-
12	Concave	Inverted	Minus-DTLZ	-
13	Convex	Triangular	DTLZ	Convex DTLZ2 [8]
14	Convex	Triangular	WFG	-
15	Convex	Triangular	Minus-DTLZ	-
16	Convex	Inverted	DTLZ	-
17	Convex	Inverted	WFG	Minus-WFG4-9 [33]
18	Convex	Inverted	Minus-DTLZ	Minus-DTLZ2-4 [33]

Now let us explain how to generate the 18 types of test problems in detail. Objective functions of our test problems with DTLZ type feasible regions in Fig. 3 (a) are written as follows:

$$\text{Minimize } f_j(\mathbf{x}) = (1 + g(\mathbf{x}_m))h_j(\mathbf{x}_{\text{pos}}), \quad j = 1, 2, \dots, m, \quad (3)$$

where $\mathbf{x} = (x_1, x_2, \dots, x_{m+k-1})$ is a decision vector with $0 \leq x_i \leq 1$ ($i = 1, 2, \dots, m+k-1$), $h_j(\mathbf{x}_{\text{pos}})$ is a position function which specifies the Pareto front shape ($h_j(\mathbf{x}_{\text{pos}}) \geq 0$), and $g(\mathbf{x}_m)$ is a distance function which specifies the distance from the Pareto front ($g(\mathbf{x}_m) \geq 0$). The decision vector $\mathbf{x}$ is divided into $\mathbf{x}_{\text{pos}} = (x_1, x_2, \dots, x_{m-1})$ in $h_j(\mathbf{x}_{\text{pos}})$ and $\mathbf{x}_m = (x_m, x_{m+1}, \dots, x_{m+k-1})$ in $g(\mathbf{x}_m)$, where k is the number of the decision variables called the distance variables in $g(\mathbf{x}_m)$. All solutions with $g(\mathbf{x}_m) = 0$ are Pareto optimal. The same formulation as (3) was used in the DTLZ test suite [29]. Eq.(3) can be rewritten as follows:

$$\text{Minimize } \mathbf{f}(\mathbf{x}) = (1 + g(\mathbf{x}_m))\mathbf{h}(\mathbf{x}_{\text{pos}}). \quad (4)$$

The Pareto front is specified by $\mathbf{z}^* = \mathbf{h}(\mathbf{x}_{\text{pos}})$ for $0 \leq x_i \leq 1$, $i = 1, 2, \dots, m-1$. From (4), $\mathbf{z}$ is a feasible solution in the objective space when the following relation holds for $\mathbf{z}^*$ on the Pareto front:

$$\mathbf{z} = (1 + g(\mathbf{x}_m))\mathbf{z}^*. \quad (5)$$

If the Pareto front is the line between (1, 0) and (0, 1) in the objective space as in Fig. 3 (a), all points on the line between (a, 0) and (0, a) are feasible solutions for $a = 1 + g(\mathbf{x}_m)$. Thus the size of

the feasible region increases with the increase of the distance from the Pareto front (i.e., with the increase of the distance function value $g(\mathbf{x}_m)$).

Objective functions of our test problems with WFG type feasible regions in Fig. 3 (b) are written as follows:

$$\text{Minimize } f_j(\mathbf{x}) = g(\mathbf{x}_m) + h_j(\mathbf{x}_{\text{pos}}), \quad j = 1, 2, \dots, m. \quad (6)$$

The difference between (3) and (6) is whether the position function $h_j(\mathbf{x}_{\text{pos}})$ is multiplied or added by the distance function $g(\mathbf{x}_m)$. This difference in their formulations leads to the difference in the shape of the feasible regions. All solutions with $g(\mathbf{x}_m) = 0$ are Pareto optimal. The same formulation as (6) was used in the WFG test suite [32].

Eq.(6) can be rewritten as follows:

$$\text{Minimize } \mathbf{f}(\mathbf{x}) = g(\mathbf{x}_m)\mathbf{1} + \mathbf{h}(\mathbf{x}_{\text{pos}}), \quad (7)$$

where $\mathbf{1} = (1, 1, \dots, 1)^T$. The Pareto front is specified by $\mathbf{z}^* = \mathbf{h}(\mathbf{x}_{\text{pos}})$ for $0 \leq x_i \leq 1$, $i = 1, 2, \dots, m-1$. That is, the two formulations in (3) and (6) have exactly the same Pareto front.

Eq.(7) shows that $\mathbf{z}$ is a feasible solution in the objective space when the following relation holds for $\mathbf{z}^*$ on the Pareto front:

$$\mathbf{z} = g(\mathbf{x}_m)\mathbf{1} + \mathbf{z}^*. \quad (8)$$

If the Pareto front is the line between (1, 0) and (0, 1) in the objective space as in Fig. 3 (b), all points on the line between (1+a, a) and (a, 1+a) are feasible solutions for $a = g(\mathbf{x}_m)$. Thus the size of the feasible region does not change with the increase of the distance from the Pareto front (i.e., with the increase of the distance function value $g(\mathbf{x}_m)$).

Our test problems with Minus-DTLZ type feasible regions are generated by multiplying a minus sign “-” to all objectives of our test problems with DTLZ type feasible regions as illustrated in Fig. 4. In [33], Minus-DTLZ and Minus-WFG test problems were generated in the same manner. After multiplying all objectives in (3) by “-”, the objective space is normalized so that the ideal and nadir points are (0, 0, ..., 0)^T and (1, 1, ..., 1)^T, respectively. It should be noted that all test problems in this paper have the same ideal point (0, 0, ..., 0)^T and the same nadir point (1, 1, ..., 1)^T.

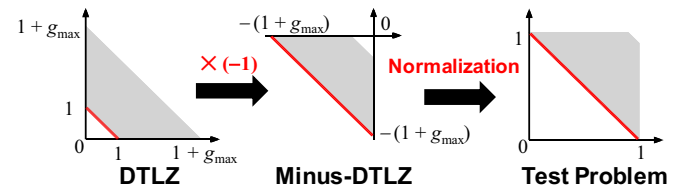


Figure 4. Illustration of our test problems with Minus-DTLZ type feasible regions.

After the normalization, objective functions of our test problems with Minus-DTLZ type feasible regions are written as follows:

$$\text{Minimize } f_j(\mathbf{x}) = 1 - \left(1 - \frac{g(\mathbf{x}_m)}{1 + g_{\text{max}}}\right)h_j(\mathbf{x}_{\text{pos}}), \quad j = 1, 2, \dots, m. \quad (9)$$

All solutions with $g(\mathbf{x}_m) = 0$ are Pareto optimal. Eq. (9) can be rewritten as follows:

$$\text{Minimize } \mathbf{f}(\mathbf{x}) = \mathbf{1} - \left(1 - \frac{g(\mathbf{x}_m)}{1 + g_{\max}}\right) \mathbf{h}(\mathbf{x}_{\text{pos}}). \quad (10)$$

The Pareto front is specified by $\mathbf{z}^* = \mathbf{1} - \mathbf{h}(\mathbf{x}_{\text{pos}})$. Note that the Pareto fronts of the test problems with the DTLZ and WFG type feasible regions are specified by $\mathbf{z}^* = \mathbf{h}(\mathbf{x}_{\text{pos}})$ in (5) and (7).

As $g(\mathbf{x}_m)$, we use the distance function in DTLZ1 [29]:

$$g(\mathbf{x}_m) = 100 \cdot \left(|\mathbf{x}_m| + \sum_{i=1}^{m-1} \{(x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5))\} \right). \quad (11)$$

Of course, we can use any distance function. In this paper, we always use the distance function in (11) for all test problems. The number of distance variables is specified as $k = 5$ as in DTLZ [29].

We generate linear, concave and convex triangular Pareto fronts using the position functions of DTLZ2 [29] in the same manner as in [35]. First we define the following m functions with $\mathbf{x}_{\text{pos}} = (x_1, x_2, \dots, x_{m-1})$ and $0 \leq x_i \leq 1$ for $i = 1, 2, \dots, m-1$:

$$\begin{aligned} s_1(\mathbf{x}_{\text{pos}}) &= [\cos(x_1\pi/2) \dots \cos(x_{m-2}\pi/2) \cos(x_{m-1}\pi/2)]^\beta, \\ s_2(\mathbf{x}_{\text{pos}}) &= [\cos(x_1\pi/2) \dots \cos(x_{m-2}\pi/2) \sin(x_{m-1}\pi/2)]^\beta, \\ &\vdots \\ s_{m-1}(\mathbf{x}_{\text{pos}}) &= [\cos(x_1\pi/2) \sin(x_2\pi/2)]^\beta, \\ s_m(\mathbf{x}_{\text{pos}}) &= [\sin(x_1\pi/2)]^\beta, \end{aligned} \quad (12)$$

where β is a positive parameter. Depending on the value of β , these functions generate triangular concave ($\beta < 2$), linear ($\beta = 2$) and convex ($\beta > 2$) Pareto fronts as shown in Fig. 5. When $\beta = 1$, these functions are the same as the position functions in DTLZ2 [29].

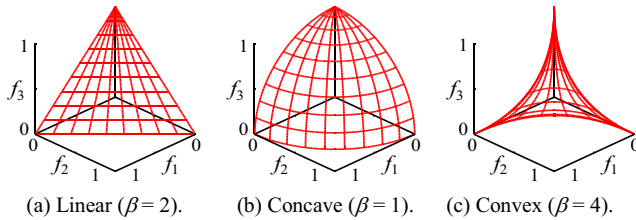


Figure 5. Triangular Pareto fronts generated by $s_j(\mathbf{x}_{\text{pos}})$ in (12).

Inverted triangular Pareto fronts can be generated from the following functions:

$$u_j(\mathbf{x}_{\text{pos}}) = 1 - s_j(\mathbf{x}_{\text{pos}}), \quad j = 1, 2, \dots, m, \quad (13)$$

where $s_j(\mathbf{x}_{\text{pos}})$ is the same as in (12). Depending on the value of β , these functions generate inverted triangular convex ($\beta < 2$), linear ($\beta = 2$) and concave ($\beta > 2$) Pareto fronts as shown in Fig. 6.

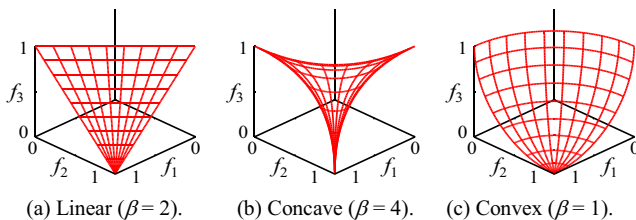


Figure 6. Inverted triangular Pareto fronts generated by $u_j(\mathbf{x}_{\text{pos}})$ in (13).

The 18 types of test problems are generated by combining the three formulations for feasible regions and the six settings for Pareto fronts as shown in Table 2. It should be noted that the Pareto front is specified by $\mathbf{z}^* = \mathbf{1} - \mathbf{h}(\mathbf{x}_{\text{pos}})$ in the Minus-DTLZ type problems whereas it is specified by $\mathbf{z}^* = \mathbf{h}(\mathbf{x}_{\text{pos}})$ in the DTLZ and WFG type problems. This difference makes Table 2 look inconsistent. For example, the type 18 test problem has the inverted convex Pareto front which is generated by $\mathbf{u}(\mathbf{x}_{\text{pos}}) = \mathbf{1} - \mathbf{s}(\mathbf{x}_{\text{pos}})$ with $\beta = 1$ as shown in Fig. 6 (c). However, since it has the Minus-DTLZ type feasible region, the Pareto front is specified by $\mathbf{z}^* = \mathbf{1} - \mathbf{h}(\mathbf{x}_{\text{pos}})$. As a result, $\mathbf{h}(\mathbf{x}_{\text{pos}})$ is defined by $\mathbf{h}(\mathbf{x}_{\text{pos}}) = \mathbf{s}(\mathbf{x}_{\text{pos}})$ for the type 18 test problem in Table 2.

Table 2. Formulations of 18 types of our test problems.

Type	Curvature	Pareto F.	Feasible R.	$\mathbf{f}(\mathbf{x})$	$\mathbf{h}(\mathbf{x}_{\text{pos}})$	β
01	Linear	Triangular	DTLZ	(4)	$\mathbf{s}(\mathbf{x}_{\text{pos}})$	2
02	Linear	Triangular	WFG	(7)	$\mathbf{s}(\mathbf{x}_{\text{pos}})$	2
03	Linear	Triangular	Minus-D.	(10)	$\mathbf{1} - \mathbf{s}(\mathbf{x}_{\text{pos}})$	2
04	Linear	Inverted	DTLZ	(4)	$\mathbf{1} - \mathbf{s}(\mathbf{x}_{\text{pos}})$	2
05	Linear	Inverted	WFG	(7)	$\mathbf{1} - \mathbf{s}(\mathbf{x}_{\text{pos}})$	2
06	Linear	Inverted	Minus-D.	(10)	$\mathbf{s}(\mathbf{x}_{\text{pos}})$	2
07	Concave	Triangular	DTLZ	(4)	$\mathbf{s}(\mathbf{x}_{\text{pos}})$	1
08	Concave	Triangular	WFG	(7)	$\mathbf{s}(\mathbf{x}_{\text{pos}})$	1
09	Concave	Triangular	Minus-D.	(10)	$\mathbf{1} - \mathbf{s}(\mathbf{x}_{\text{pos}})$	1
10	Concave	Inverted	DTLZ	(4)	$\mathbf{1} - \mathbf{s}(\mathbf{x}_{\text{pos}})$	4
11	Concave	Inverted	WFG	(7)	$\mathbf{1} - \mathbf{s}(\mathbf{x}_{\text{pos}})$	4
12	Concave	Inverted	Minus-D.	(10)	$\mathbf{s}(\mathbf{x}_{\text{pos}})$	4
13	Convex	Triangular	DTLZ	(4)	$\mathbf{s}(\mathbf{x}_{\text{pos}})$	4
14	Convex	Triangular	WFG	(7)	$\mathbf{s}(\mathbf{x}_{\text{pos}})$	4
15	Convex	Triangular	Minus-D.	(10)	$\mathbf{1} - \mathbf{s}(\mathbf{x}_{\text{pos}})$	4
16	Convex	Inverted	DTLZ	(4)	$\mathbf{1} - \mathbf{s}(\mathbf{x}_{\text{pos}})$	1
17	Convex	Inverted	WFG	(7)	$\mathbf{1} - \mathbf{s}(\mathbf{x}_{\text{pos}})$	1
18	Convex	Inverted	Minus-D.	(10)	$\mathbf{s}(\mathbf{x}_{\text{pos}})$	1

3 COMPUTATIONAL EXPERIMENTS

Using the generated 18 types of test problems with 3, 5, 8 and 10 objectives ($m = 3, 5, 8, 10$), we compare NSGA-II with the following well-known decomposition-based EMO algorithms: MOEA/D [6], NSGA-III [8], MOEA/DD [9] and θ -DEA [10]. In MOEA/D, we use the same scalarizing function as in the MOEA/D paper [6] for DTLZ1-2: the PBI function with $\theta = 5$. The following settings are used in all the examined EMO algorithms:

Population size: 105 ($m = 3$), 210 ($m = 5$), 156 ($m = 8$), 275 ($m = 10$).

Termination conditions: 1000 generations.

Crossover: SBX (Probability 1.0, Index 20).

Mutation: PM (Probability $1/n$, Index 20).

The same or similar settings were often used in recent studies on evolutionary many-objective optimization [8]–[16]. When the decomposition-based EMO algorithms are applied to test problems with 3 and 5 objectives, all weight vectors $\mathbf{w} = (w_1, w_2, \dots, w_m)^T$ satisfying the following relations are generated as in the MOEA/D paper [6]:

$$w_1 + w_2 + \dots + w_m = 1, \quad (14)$$

$$w_i \in \left\{0, \frac{1}{H}, \frac{2}{H}, \dots, \frac{H}{H}\right\}, \quad (15)$$

where H is an integer parameter which defines the number of weight vectors ($H = 13$ for $m = 3$ and $H = 6$ for $m = 5$ in this paper).

For test problems with 8 and 10 objectives, the two-layered method in NSGA-III [8] is used to generate weight vectors in all the four decomposition-based EMO algorithms in this paper.

As explained in Section 2, the DTLZ1 distance function in (11) is used in all test problems in this paper. The number of distance variables is specified as $k = 5$. Thus the total number of decision variables n is $n = m + 4$. Each variable has the same box constraint: $0 \leq x_i \leq 1$ for $i = 1, 2, \dots, n$.

Each EMO algorithm is applied to each test problem 51 times. The obtained solution set from each run is evaluated by the HV indicator [20], IGD indicator [21], and IGD⁺ indicator [36]. Since HV-based comparison results depend on the specification of the reference point $\mathbf{r} = (r, r, \dots, r)^T$ for HV calculation, we examine its two specifications: $\mathbf{r} = (1.1, 1.1, \dots, 1.1)^T$ and $\mathbf{r} = (1+1/H, 1+1/H, \dots, 1+1/H)^T$ where H is the integer parameter used for generating the weight vectors in MOEA/D. More specifically, $H = 13$ ($m = 3$), $H = 6$ ($m = 5$), $H = 3$ ($m = 8$), and $H = 3$ ($m = 10$). This specification was proposed for fair performance comparison in [37]. Due to the page limitation as a conference paper, we mainly report the HV-based comparison results with the second specification: $\mathbf{r} = 1+1/H$. All experimental results are available from the authors' webpage together with other related materials such as the code of each algorithm, explanations about each test problem, and a reference point set for each test problem for IGD and IGD⁺ (see <http://www.cs.osakafu-u.ac.jp/ci/EnglishPage/Download.php>). We run all experiments in MATLAB using PlatEMO [38].

HV-based comparison results ($\mathbf{r} = 1+1/H$) on the 8-objective and 10-objective test problems are summarized in Tables 3 and 4, respectively. Each algorithm is compared with NSGA-II using a statistical test at the 5% significance level (Wilcoxon's rank sum test, Welch's t-test, or Student's t-test depending on the property of the distribution of HV values obtained by each algorithm). In each table, statistical test results are shown by "+" and "-", which mean that the compared results are significantly better than and worse than those of NSGA-II, respectively. When there is no significant difference, "=" is assigned. The number of "+", "=", and "-" is counted for each algorithm at the bottom of each table.

Table 3. Average HV values ($\mathbf{r} = 1 + 1/H$) on the 8-objective test problems. The best and worst results are shown by red and blue fonts, respectively. The statistical comparison results are summarized in the last row.

Type	MOEA/DD	MOEA/D	NSGA-II	NSGA-III	θ -DEA
01	9.98e+00 (+)	9.98e+00 (+)	0.00e+00	9.95e+00 (+)	9.97e+00 (+)
02	9.96e+00 (+)	9.93e+00 (=)	9.94e+00	9.96e+00 (+)	9.97e+00 (+)
03	9.98e+00 (+)	9.98e+00 (+)	9.91e+00	9.98e+00 (+)	9.98e+00 (+)
04	0.66e-02 (-)	0.46e-02 (-)	1.80e-02	1.17e-02 (-)	0.91e-02 (-)
05	0.61e-02 (-)	0.50e-02 (-)	1.93e-02	1.19e-02 (-)	0.93e-02 (-)
06	0.61e-02 (-)	0.66e-02 (-)	1.91e-02	1.17e-02 (-)	0.99e-02 (-)
07	9.82e+00 (+)	9.77e+00 (+)	0.00e+00	9.74e+00 (+)	9.77e+00 (+)
08	9.80e+00 (+)	9.68e+00 (+)	9.46e+00	9.60e+00 (+)	9.79e+00 (+)
09	9.82e+00 (+)	9.81e+00 (+)	9.46e+00	9.79e+00 (+)	9.82e+00 (+)
10	1.84e-03 (-)	0.80e-03 (-)	6.22e-03	1.98e-03 (-)	1.39e-03 (-)
11	1.78e-03 (-)	0.74e-03 (-)	6.19e-03	2.15e-03 (-)	2.06e-03 (-)
12	1.78e-03 (-)	2.18e-03 (-)	5.67e-03	3.17e-03 (-)	1.93e-03 (-)
13	9.78e+00 (+)	9.74e+00 (+)	0.00e+00	9.98e+00 (+)	9.97e+00 (+)
14	9.78e+00 (-)	9.72e+00 (-)	9.99e+00	9.97e+00 (-)	9.96e+00 (-)
15	9.80e+00 (-)	9.74e+00 (-)	9.99e+00	9.98e+00 (-)	9.97e+00 (-)
16	1.15e-01 (-)	1.30e-01 (+)	1.21e-01	1.86e-01 (+)	1.45e-01 (+)
17	1.18e-01 (-)	1.30e-01 (-)	1.67e-01	1.85e-01 (+)	1.43e-01 (-)
18	1.14e-01 (-)	1.31e-01 (-)	1.87e-01	1.65e-01 (-)	1.44e-01 (-)
+/-/-	7/0/11	7/1/10	-	9/0/9	8/0/10

Table 4. Average HV values ($\mathbf{r} = 1 + 1/H$) on the 10-objective test problems. The best and worst results are shown by red and blue fonts, respectively. The statistical comparison results are summarized in the last row.

Type	MOEA/DD	MOEA/D	NSGA-II	NSGA-III	θ -DEA
01	1.77e+01 (+)	1.77e+01 (+)	0.00e+00	1.77e+01 (+)	1.77e+01 (+)
02	1.77e+01 (+)	1.77e+01 (=)	1.77e+01	1.77e+01 (+)	1.77e+01 (+)
03	1.77e+01 (+)	1.77e+01 (+)	1.77e+01	1.77e+01 (+)	1.77e+01 (+)
04	0.73e-03 (-)	0.48e-03 (-)	3.06e-03	1.78e-03 (-)	1.28e-03 (-)
05	0.66e-03 (-)	0.52e-03 (-)	3.41e-03	1.81e-03 (-)	1.21e-03 (-)
06	0.54e-03 (-)	0.73e-03 (-)	3.60e-03	1.67e-03 (-)	1.01e-03 (-)
07	1.76e+01 (+)	1.74e+01 (+)	0.00e+00	1.74e+01 (+)	1.76e+01 (+)
08	1.76e+01 (+)	1.74e+01 (+)	1.72e+01	1.75e+01 (+)	1.76e+01 (+)
09	1.76e+01 (+)	1.76e+01 (+)	1.72e+01	1.76e+01 (+)	1.76e+01 (+)
10	2.22e-04 (-)	0.94e-04 (-)	9.57e-04	2.11e-04 (-)	1.65e-04 (-)
11	2.13e-04 (-)	0.77e-04 (-)	9.54e-04	2.21e-04 (-)	2.08e-04 (-)
12	2.11e-04 (-)	2.69e-04 (-)	8.71e-04	3.97e-04 (-)	2.23e-04 (-)
13	1.73e+01 (+)	1.72e+01 (+)	0.00e+00	1.77e+01 (+)	1.77e+01 (+)
14	1.73e+01 (-)	1.72e+01 (-)	1.77e+01	1.77e+01 (-)	1.77e+01 (-)
15	1.74e+01 (-)	1.72e+01 (-)	1.77e+01	1.77e+01 (-)	1.77e+01 (-)
16	2.37e-02 (-)	3.52e-02 (=)	3.61e-02	6.78e-02 (+)	5.11e-02 (+)
17	2.50e-02 (-)	3.54e-02 (-)	5.13e-02	6.98e-02 (+)	5.04e-02 (=)
18	2.44e-02 (-)	3.56e-02 (-)	6.64e-02	6.07e-02 (-)	4.86e-02 (-)
+/-/-	7/0/11	6/2/10	-	9/0/9	8/1/9

In Tables 3 and 4, severe performance deterioration of NSGA-II is observed for the following three types (see the blue font results):

Severe Performance Deterioration of NSGA-II

Type 01: Linear, triangular, DTLZ type.

Type 07: Concave, triangular, DTLZ type.

Type 13: Convex, triangular, DTLZ type.

Independent of the curvature property, many-objective DTLZ type test problems with triangular Pareto fronts are very difficult for NSGA-II. For those test problems with 8 and 10 objectives, the average HV values are always zero in Tables 3 and 4. This means that no solutions which dominate the reference point are obtained. The observed poor performance of NSGA-II on the three types (01, 07 and 13) in Tables 3 and 4 is consistent with repeatedly reported severe performance deterioration of NSGA-II on many-objective DTLZ test problems [26], [28]. As shown in Table 1, DTLZ1 and DTLZ2-4 are categorized as type 01 and type 07, respectively.

Surprisingly, the best results highlighted by the red font are obtained by NSGA-II in both tables for the following problems:

Best Results by NSGA-II

Type 04: Linear, inverted triangular, DTLZ type

Type 05: Linear, inverted triangular, WFG type

Type 06: Linear, inverted triangular, Minus-DTLZ type

Type 10: Concave, inverted triangular, DTLZ type

Type 11: Concave, inverted triangular, WFG type

Type 12: Concave, inverted triangular, Minus-DTLZ type

Type 14: Convex, triangular, WFG type.

Type 15: Convex, triangular, Minus-DTLZ type

Type 18: Convex, inverted triangular, Minus-DTLZ type

The best results are obtained by NSGA-II for these 9 (out of 18) types in both tables. This is a surprising observation since poor performance of NSGA-II on many-objective problems has been repeatedly reported in the literature. However, we can also see from Table 1 that these nine types of test problems have not been frequently used for the performance evaluation of many-objective algorithms in the literature. DTLZ1 (type 01), DTLZ2-4 (type 07)

and WFG4-9 (type 08) have been used in many studies [5]–[16], [26], [28]. In Tables 3 and 4, the worst results are always obtained by NSGA-II for these three types (types 01, 07, 08) of test problems. These observations suggest that algorithm comparison has been performed in the last decade using many-objective test problems which are difficult for NSGA-II.

Different comparison results are often obtained from different performance indicators. We also count the number of 10-objective problems for which the best results are obtained by NSGA-II based on other performance indicators. Performance comparison results are summarized as follows: The best results are obtained by NSGA-II for 5 types (HV with $r = 1.1$), 6 types (IGD), and 8 types (IGD⁺) out of the 18 types of test problems. When $r = 1 + 1/H$ (i.e., $r = 1 + 1/3$ for $m = 10$), a well-distributed solution set over the entire Pareto front including its boundary region usually has a high HV value. By decreasing the distance from the nadir point to the reference point (i.e., by decreasing the value of r), we can give higher importance to inside solutions than boundary solutions [37]. As a result, the evaluation of NSGA-II becomes weaker by changing the reference point specification from $r = 1 + 1/H$ to $r = 1.1$ (since NSGA-II does not have high convergence ability for many-objective problems whereas it has high diversification ability). However, even when r is specified as $r = 1.1$ for HV calculation, the best results are obtained by NSGA-II for 5 (out of 18) types of test problems. This observation shows high overall performance of NSGA-II for some many-objective test problems.

In order to further examine this issue, we calculate the average distance function value for all solutions in the final population over 51 runs of each EMO algorithm on each 10-objective test problem. Experimental results are summarized in Table 5.

Table 5. Average distance function value on the 10-objective test problems. The best and worst results are shown by red and blue fonts, respectively. The statistical comparison results are summarized in the last row.

Type	MOEA/DD	MOEA/D	NSGA-II	NSGA-III	θ -DEA
01	0.001 (+)	0.003 (+)	1007.489	3.826 (+)	0.169 (+)
02	0.000 (+)	0.001 (+)	0.061	0.000 (+)	0.000 (+)
03	1.814 (+)	0.413 (+)	81.961	2.152 (+)	0.198 (+)
04	0.000 (+)	0.001 (+)	0.046	0.000 (+)	0.000 (+)
05	0.000 (+)	0.000 (+)	0.022	0.000 (+)	0.000 (+)
06	0.206 (+)	0.000 (+)	69.799	53.317 (+)	2.410 (+)
07	0.000 (+)	0.001 (+)	1037.140	3.000 (+)	0.460 (+)
08	0.001 (+)	0.004 (+)	0.118	0.000 (+)	0.000 (+)
09	23.429 (+)	24.241 (+)	181.540	11.918 (+)	11.503 (+)
10	0.089 (–)	0.001 (+)	0.003	0.001 (+)	0.042 (–)
11	0.000 (+)	0.000 (+)	0.002	0.001 (+)	0.000 (+)
12	4.762 (+)	0.000 (+)	122.980	62.331 (+)	0.977 (+)
13	0.001 (+)	0.002 (+)	915.301	15.939 (+)	1.935 (+)
14	0.000 (+)	0.000 (+)	0.002	0.000 (+)	0.000 (+)
15	0.005 (+)	0.007 (+)	6.350	0.110 (+)	0.016 (+)
16	0.000 (+)	0.000 (+)	0.146	0.002 (+)	0.001 (+)
17	0.000 (+)	0.000 (+)	0.043	0.001 (+)	0.001 (+)
18	0.068 (+)	0.045 (+)	41.477	64.802 (–)	7.984 (+)
+ / – / =	17/0/1	18/0/0	–	17/0/1	17/0/1

As we have already explained, solutions are Pareto optimal when the distance function values are zero. Small values of the distance function mean that solutions are close to the Pareto front. In Table 5, the worst results are obtained by NSGA-II for 16 (out of 18) types of test problems. This observation clearly shows that the convergence ability of NSGA-II is weak for many-objective

problems. However, the overall performance of NSGA-II evaluated by the HV, IGD and IGD+ indicators is the best for some test problems. These observations imply that the obtained solution sets by NSGA-II have larger diversity than those by the decomposition-based algorithms examined in this paper.

4 DISCUSSIONS

First, we discuss why the performance of NSGA-II on the DTLZ type test problems with triangular Pareto fronts (i.e., types 01, 07, 13) severely degrades with the increase in the number of objectives. In Fig. 7, we show the feasible region of the type 01 (linear, triangular, DTLZ) test problem with three objectives. All the three DTLZ type test problems with triangular Pareto fronts (i.e., types 01, 07, 13) have similar feasible regions.

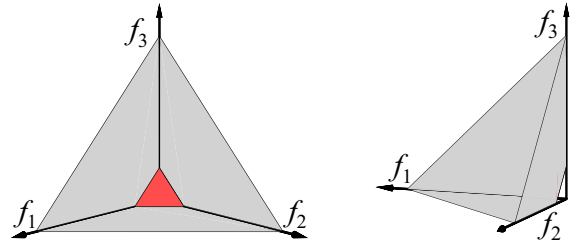


Figure 7. Feasible region of the type 01 test problem (red: Pareto front).

As we can see from Fig. 7, the type 01 (and also types 07, 13) test problem has the feasible region along each axis of the objective space. For example, a point $(10^{-6}, 10^{-6}, 100)$ is included in the feasible region. This solution is far away from the Pareto front of the type 01 test problem since it is the plane specified by $f_1 + f_2 + f_3 = 1$ and $0 \leq f_i \leq 1$ for $i = 1, 2, 3$. Whereas $(10^{-6}, 10^{-6}, 100)$ is not a good solution, it is not easy for NSGA-II to find a better solution satisfying $0 \leq f_1 \leq 10^{-6}$, $0 \leq f_2 \leq 10^{-6}$ and $0 \leq f_3 \leq 100$. This difficulty becomes very severe by increasing the number of objectives. For example, it is very difficult to find a solution with $0 \leq f_i \leq 10^{-6}$ for $i = 1, 2, \dots, 9$ and $0 \leq f_{10} \leq 100$ for the 10-objective type 01 test problem. As a result, such a solution can remain in the population over many generations. Moreover, similar non-dominated solutions such as $(0.5 \times 10^{-6}, 2 \times 10^{-6}, 50)$, $(0.6 \times 10^{-6}, 1.1 \times 10^{-6}, 150)$ and $(3 \times 10^{-6}, 0.2 \times 10^{-6}, 200)$ can be generated. All of them are non-dominated, and they are likely to be evaluated as being better than solutions around the Pareto front by the crowding distance. Those solutions close to each axis of the objective space (and far away from the Pareto front) are called dominance resistant solutions (DRSs [39]). It is well known that the existence of DRSs severely degrades the performance of Pareto dominance-based EMO algorithms [39], [40]. When the shape of the feasible regions is WFG type or Minus-DTLZ type, each axis of the objective space is not included in the feasible region (see Fig. 3). Thus no DRSs exist in the WFG and Minus-DTLZ type test problems. Even when the shape of feasible regions is DTLZ type, no DRSs exist in test problems with inverted triangular Pareto fronts. This is because no points on the axes of the objective space are feasible.

These discussions explain why the performance of NSGA-II severely degrades only on the three DTLZ type many-objective test problems with triangular Pareto fronts (i.e., types 01, 07, 13). In order to further discuss this issue, we perform computational experiments after slightly changing the objective function vector in (4) of the DTLZ type test problems as follows:

$$\text{Minimize } \mathbf{f}(\mathbf{x}) = (1 + g(\mathbf{x}_m))\mathbf{h}(\mathbf{x}_{\text{pos}}) + \alpha g(\mathbf{x}_m)\mathbf{1}, \quad (16)$$

where α is a small positive real number. When $\alpha = 0$, this objective function vector is the same as (4). By increasing the value of α , the shape of the feasible region gradually changes from the DTLZ type to the WFG type. We apply NSGA-II to the type 01 test problem with (16) using various settings of α : $\alpha = 1, 10^{-1}, 10^{-2}, \dots, 10^{-16}, 0$.

Experimental results on the type 01 test problems with 8 and 10 objectives are shown in Figs. 8 and 9, respectively. These figures show the average distance function values. In these figures, NSGA-II is compared with random search. In the random search, a pre-specified number of solutions are randomly generated. The number of examined solutions is exactly the same between NSGA-II and the random search. All non-dominated solutions are found from the examined solutions in the random search, and a final population is randomly selected from the non-dominated solutions. In Figs. 8-9, the convergence ability of NSGA-II is weaker than that of the random search when $\alpha = 0$. However, a small change of the feasible region shape by (16) using a small value of α significantly improves the convergence ability of NSGA-II.

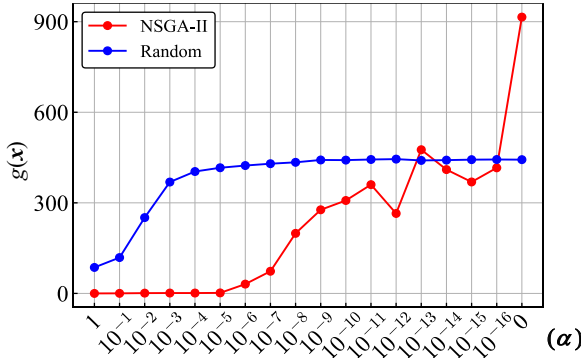


Figure 8. Average distance function values of the 8-objective type 01 test problem with the modified objective vector in (16).

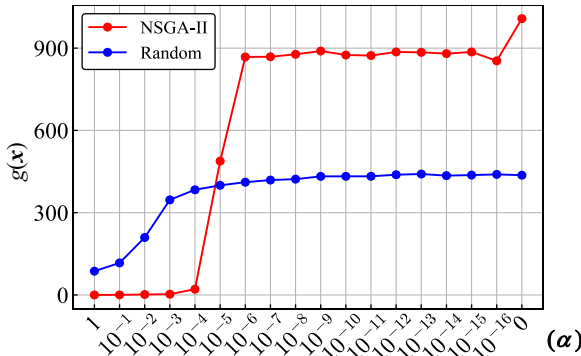


Figure 9. Average distance function values of the 10-objective type 01 test problem with the modified objective vector in (16).

For an m -objective DTLZ type test problem with a triangular Pareto front, we can simultaneously optimize $(m - 1)$ objectives. As a result, we have DRSs such as $(10^{-6}, 10^{-6}, \dots, 10^{-6}, 100)$. However, this property (i.e., the existence of solutions which can simultaneously optimize $(m - 1)$ objectives) looks unrealistic [41]. DTLZ type test problems with triangular Pareto fronts seem to be unnecessarily difficult for NSGA-II due to this unrealistic property.

Next, let us discuss why NSGA-II works well on some other many-objective test problems. As explained in [33], all objective values of a current solution can be simultaneously improved by

decreasing the distance function value when the objective vector is defined by (4), (7) and (10). Let us denote the current solution in an m -dimensional objective space be $\mathbf{z}$. If we generate a new point $\mathbf{y}$ randomly in the neighborhood of $\mathbf{z}$ in the objective space, $\mathbf{y}$ and $\mathbf{z}$ are non-dominated with each other with the probability of $1 - 2/2^m$ (since $\mathbf{y}$ dominates $\mathbf{z}$ with the probability of $1/2^m$, and $\mathbf{z}$ dominates $\mathbf{y}$ with the same probability). In the case of $m=10$, $1 - 2/2^m = 511/512$. Thus, it is likely that $\mathbf{y}$ and $\mathbf{z}$ are non-dominated with each other. However, if $\mathbf{y}$ is generated by applying a mutation operator only to distance variables, the distance function value is increased or decreased by the mutation. When the distance function value is decreased, $\mathbf{y}$ dominates $\mathbf{z}$. When the distance function value is increased, $\mathbf{y}$ is dominated by $\mathbf{z}$. Thus the probability of generating $\mathbf{y}$ which is non-dominated with $\mathbf{z}$ is zero. This explanation suggests that the special structure of our test problems (and the special structure of almost all of the frequently-used many-objective test problems such as DTLZ and WFG) helps NSGA-II to search for better solutions using the Pareto dominance relation.

5 CONCLUSION

In this paper, we examined the performance of NSGA-II through computational experiments on 18 types of test problems. NSGA-II was compared with four well-known decomposition-based EMO algorithms: MOEA/D, NSGA-III, MOEA/DD and θ -DEA. We obtained the following observations:

- (1) The performance of NSGA-II severely deteriorated with the increase in the number of objectives for three types of test problems (types 01, 07, 13). These types include DTLZ1-4.
- (2) NSGA-II outperformed the other examined EMO algorithms on a half of the 18 types of test problems with 8 and 10 objectives.

We explained the reason for the poor performance of NSGA-II, which is the existence of DRSs in the DTLZ type test problems with triangular Pareto fronts. It was shown that the convergence ability of NSGA-II on such a test problem with 10 objectives was inferior to random search. It was also shown that a slight change of the DTLZ type feasible region clearly improved the convergence ability of NSGA-II. We also discussed why NSGA-II worked well on some test problems. The reason is the special structure of the objective functions in the DTLZ and WFG test problems.

Our experimental results clearly demonstrated the necessity of including NSGA-II in computational experiments for performance evaluation of new many-objective algorithms. The necessity of good many-objective test problems was also demonstrated by our computational experiments and related discussions in this paper. One important future research issue is to create a set of reliable many-objective test problems for fair performance evaluation of existing and new many-objective algorithms.

ACKNOWLEDGEMENTS

This work was supported by National Natural Science Foundation of China (Grant No. 61876075), the Program for Guangdong Introducing Innovative and Entrepreneurial Teams (Grant No. 2017ZT07X386), Shenzhen Peacock Plan (Grant No. KQTD2016112514355531), the Science and Technology Innovation Committee Foundation of Shenzhen (Grant No. ZDSYS201703031748284), and the Program for University Key Laboratory of Guangdong Province (Grant No. 2017KSYS008).

REFERENCES

- [1] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, Chichester (2001).
- [2] H. Ishibuchi, H. Masuda, and Y. Nojima, 'Pareto fronts of many-objective degenerate test problems,' *IEEE Trans. on Evolutionary Computation*, **20**(5), 807-813 (2016).
- [3] E. Zitzler and L. Thiele, 'Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach,' *IEEE Trans. on Evolutionary Computation*, **3**(4), 257-271 (1999).
- [4] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, 'A fast and elitist multiobjective genetic algorithm: NSGA-II,' *IEEE Trans. on Evolutionary Computation*, **6**(2), 182-197 (2002).
- [5] N. Beume, B. Naujoks, and M. Emmerich, 'SMS-EMOA: Multiobjective selection based on dominated hypervolume,' *European J. of Operational Research*, **181**(3), 1653-1669 (2007).
- [6] Q. Zhang and H. Li, 'MOEA/D: A multiobjective evolutionary algorithm based on decomposition,' *IEEE Trans. on Evolutionary Computation*, **11**(6), 712-731 (2007).
- [7] J. Bader and E. Zitzler, 'HypE: An algorithm for fast hypervolume-based many-objective optimization,' *Evolutionary Computation*, **19**(1), 45-76 (2011).
- [8] K. Deb and H. Jain, 'An evolutionary many-objective optimization algorithm using reference-point-based non-dominated sorting approach, Part I: Solving problems with box constraints,' *IEEE Trans. on Evolutionary Computation*, **18**(4), 577-601 (2014).
- [9] K. Li, K. Deb, Q. Zhang, and S. Kwong, 'An evolutionary many-objective optimization algorithm based on dominance and decomposition,' *IEEE Trans. on Evolutionary Computation*, **19**(5), 694-716 (2015).
- [10] Y. Yuan, H. Xu, B. Wang, and X. Yao, 'A new dominance relation based evolutionary algorithm for many-objective optimization,' *IEEE Trans. on Evolutionary Computation*, **20**(1), 16-37 (2016).
- [11] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, 'A reference vector guided evolutionary algorithm for many-objective optimization,' *IEEE Trans. on Evolutionary Computation*, **20**(5), 773-791 (2016).
- [12] Y. Xiang, Y. Zhou, M. Li, and Z. Chen, 'A vector angle-based evolutionary algorithm for unconstrained many-objective optimization,' *IEEE Trans. on Evolutionary Computation*, **21**(1), 131-152 (2017).
- [13] R. Wang, Z. Zhou, H. Ishibuchi, T. Liao, and T. Zhang, 'Localized weighted sum method for many-objective optimization,' *IEEE Trans. on Evolutionary Computation*, **22**(1), 3-18 (2018).
- [14] X. He, Y. Zhou, Z. Chen, and Q. Zhang, 'Evolutionary many-objective optimization based on dynamical decomposition,' *IEEE Trans. on Evolutionary Computation*, **23**(3), 361-375 (2019).
- [15] Y. Sun, G. G. Yen, and Z. Yi, 'IGD indicator-based evolutionary algorithm for many-objective optimization problems,' *IEEE Trans. on Evolutionary Computation*, **23**(2), 173-187 (2019).
- [16] Y. Xiang, Y. Zhou, X. Yang, and H. Huang, 'A many-objective evolutionary algorithm with Pareto-adaptive reference points,' *IEEE Trans. on Evolutionary Computation*, **24**(1), 99-113 (2020).
- [17] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. Fonseca, 'Performance assessment of multiobjective optimizers: An analysis and review,' *IEEE Trans. on Evolutionary Computation*, **7**(2), 117-132 (2003).
- [18] O. Schütze, X. Esquivel, A. Lara, and C. A. C. Coello, 'Using the averaged Hausdorff distance as a performance measure in evolutionary multiobjective optimization,' *IEEE Trans. on Evolutionary Computation*, **16**(4) 504-522 (2012).
- [19] M. Li and X. Yao, 'Quality evaluation of solution sets in multiobjective optimisation: A survey,' *ACM Computing Surveys*, **52**(2), 1-38 (2019).
- [20] E. Zitzler and L. Thiele, 'Multiobjective optimization using evolutionary algorithms – A comparative case study,' *Proc. of PPSN V*, 292-301 (1998).
- [21] C. A. C. Coello and M. R. Sierra, 'A study of the parallelization of a coevolutionary multi-objective evolutionary algorithm,' *Lecture Notes in Computer Science 2972: MICAI 2004*, 688-697 (2004).
- [22] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, 'Evolutionary many-objective optimization: A short review,' *Proc. of IEEE CEC 2008*, 2424-2431 (2008).
- [23] C. von Lücken, B. Barán, and C. Brizuela, 'A survey on multi-objective evolutionary algorithms for many-objective problems,' *Computational Optimization and Applications*, **58**(3), 707-756 (2014).
- [24] B. Li, J. Li, K. Tang, and X. Yao, 'Many-objective evolutionary algorithms: A survey,' *ACM Computing Surveys*, **48**(1), 1-35 (2015).
- [25] H. Sato, H. E. Aguirre, and K. Tanaka, 'Controlling dominance area of solutions and its impact on the performance of MOEAs,' *Lecture Notes in Computer Science 4403: EMO 2007*, 5-20 (2007).
- [26] T. Wagner, N. Beume, and B. Naujoks, 'Pareto-, aggregation-, and indicator-based methods in many-objective optimization,' *Lecture Notes in Computer Science 4403: EMO 2007*, 742-756 (2007).
- [27] D. Corne and J. Knowles, 'Techniques for highly multiobjective optimisation: Some nondominated points are better than others,' *Proc. of GECCO 2007*, 773-780 (2007).
- [28] S. Mostaghim and H. Schmeck, 'Distance based ranking in many-objective particle swarm optimization,' *Proc. of PPSN X*, 753-762 (2008).
- [29] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, 'Scalable multi-objective optimization test problems,' *Proc. of IEEE CEC 2002*, 825-830 (2002).
- [30] O. Schütze, A. Lara, and C. A. C. Coello, 'On the influence of the number of objectives on the hardness of a multiobjective optimization problem,' *IEEE Trans. on Evolutionary Computation*, **15**(4), 444-455 (2011).
- [31] H. Ishibuchi, N. Akedo, and Y. Nojima, 'Behavior of multi-objective evolutionary algorithms on many-objective knapsack problems,' *IEEE Trans. on Evolutionary Computation*, **19**(2), 264-283 (2015).
- [32] S. Huband, P. Hingston, L. Barone, and L. While, 'A review of multiobjective test problems and a scalable test problem toolkit,' *IEEE Trans. on Evolutionary Computation*, **10**(5), 477-506 (2006).
- [33] H. Ishibuchi, Y. Setoguchi, H. Masuda, and Y. Nojima, 'Performance of decomposition-based many-objective algorithms strongly depends on Pareto front shapes,' *IEEE Trans. on Evolutionary Computation*, **21**(2), 169-190 (2017).
- [34] H. Jain and K. Deb, 'An improved adaptive approach for elitist nondominated sorting genetic algorithm for many-objective optimization,' *Lecture Notes in Computer Science 7811: EMO 2013*, 307-321 (2013).
- [35] H. Ishibuchi, R. Imada, N. Masuyama, and Y. Nojima, 'Two-layered weight vector specification in decomposition-based multi-objective algorithms for many-objective optimization problems,' *Proc. of IEEE CEC 2019*, 2435-2442 (2019).
- [36] H. Ishibuchi, H. Masuda, Y. Tanigaki, and Y. Nojima, 'Modified distance calculation in generational distance and inverted generational distance,' *Lecture Notes in Computer Science 9018: EMO 2015*, 110-125 (2015).
- [37] H. Ishibuchi, R. Imada, Y. Setoguchi, and Y. Nojima, 'How to specify a reference point in hypervolume calculation for fair performance comparison,' *Evolutionary Computation*, **26**(3), 411-440 (2018).
- [38] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, 'PlatEMO: A MATLAB platform for evolutionary multi-objective optimization,' *IEEE Computational Intelligence Magazine*, **12**(4), 73-87 (2017).
- [39] K. Ikeda, H. Kita, and S. Kobayashi, 'Failure of Pareto-based MOEAs: Does non-dominated really mean near to optimal?' *Proc. of IEEE CEC 2001*, 957-962 (2001).
- [40] Z. Wang, Y. S. Ong, and H. Ishibuchi, 'On scalable multiobjective test problems with hardly-dominated boundaries,' *IEEE Trans. on Evolutionary Computation*, **23**(2), 217-231 (2019).
- [41] H. Ishibuchi, L. He, and K. Shang, 'Regular Pareto front shape is not realistic,' *Proc. of IEEE CEC 2019*, 2035-2042 (2019).